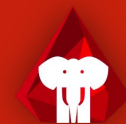




Rails Girls Nantes

Samedi 26 septembre 2020



Un événement proposé par Nantes.rb



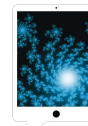
le wagon
Nantes



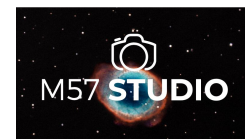
clever cloud



Code Lutin



La Voix du Chat Artiste





Cécile Veneziani

Freelance dev web

Parle le Ruby 💎 depuis 2008
et Néerlandais 🇳🇱 depuis peu

@cecilitse sur Twitter

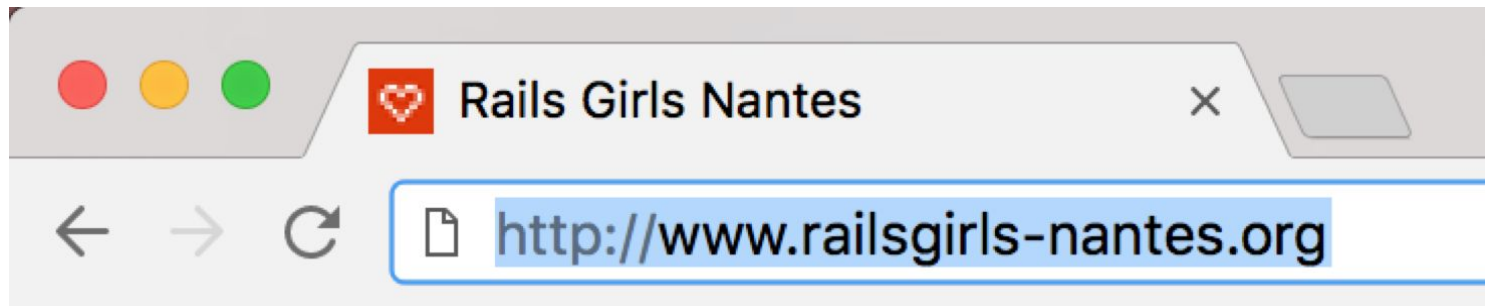


Co-driver au Wagon Nantes & Rennes

@lewagonnantes
Instagram / Twitter / LinkedIn

COMMENT FONCTIONNE LE **WEB** ?

CE QUI SE PASSE QUAND ON SE REND SUR UN SITE WEB





1

GET / HTTP/1.1

Host: www.railsgirls-nantes.org

Internet

3

*Envoi des ressources
demandées :*

Page HTML

Feuilles de styles (CSS)

Javascript

Son / Image / Vidéo

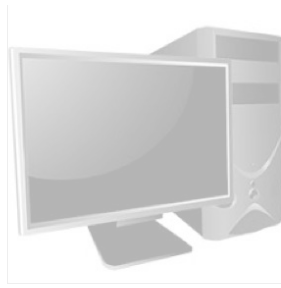
PDF

...

2

Traitement de la requête





1

GET / HTTP/1.1

Host: www.railsgirls-nantes.org



Internet

*Envoi des ressources
demandées :*

Page HTML

Feuilles de styles (CSS)

Javascript

Son / Image / Vidéo

PDF

...

3



2

Application Web



APPRENDRE À **PROGRAMMER** POUR LE **WEB**

UNE INTRODUCTION À LA PROGRAMMATION WEB AVEC



FRAMEWORK DE DÉVELOPPEMENT WEB

FRAMEWORK =



HTTP

PROGRAMMATION ORIENTÉE OBJET

PARAMÈTRES

VUE

RUBY

BASE DE DONNÉES

MODÈLE

SQL

FRAMEWORK

RAILS

CSS

ERB

MVC

CONTRÔLEUR

REQUÊTE

REST

JAVASCRIPT

HTML



RAILS EST MAGIQUE ?

Pas besoin de tout comprendre
pour commencer à construire
une application web.

RUBY ON RAILS EST
UN FRAMEWORK WEB
GÉNÉRALISTE

E-COMMERCE
MARKET PLACES
APPLICATIONS DE GESTION
API
RESEAUX SOCIAUX
...

Ce que vous voulez !



GROUPON[®]

Etsy



KICKSTARTER





+

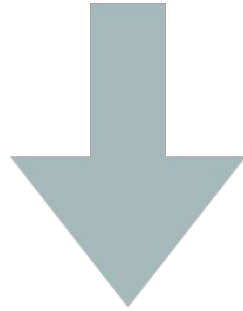
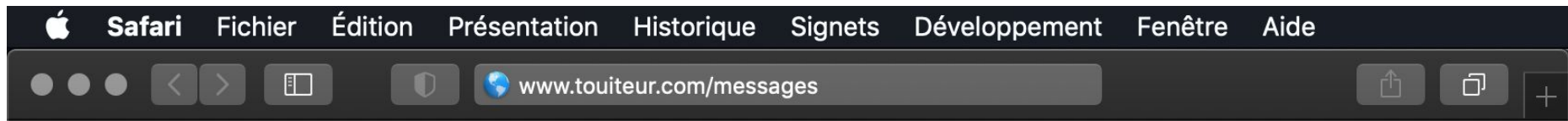


=

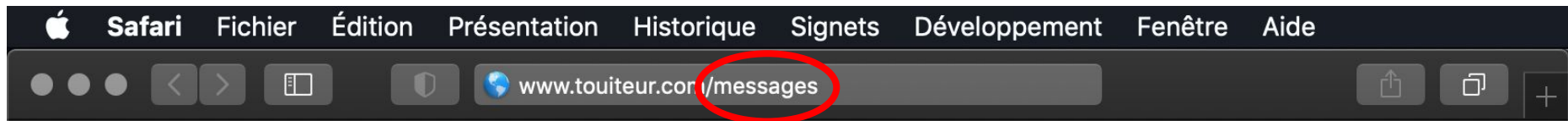


LE WEB EN MODE MVC





GET /messages HTTP/1.1
Host: www.touiteur.com



GET /messages HTTP/1.1
Host: www.touiteur.com

« Donne-moi la page
HTML qui affiche la
liste des messages »



1

GET /messages HTTP/1.1
Host: www.touiteur.com

Internet

3

2

Traitement de la requête



*Envoi des ressources
demandées :*

Page HTML

Feuilles de styles (CSS)

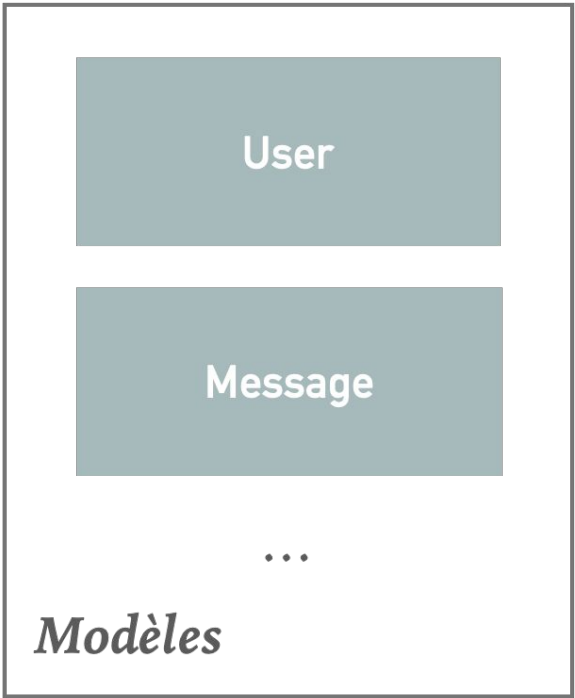
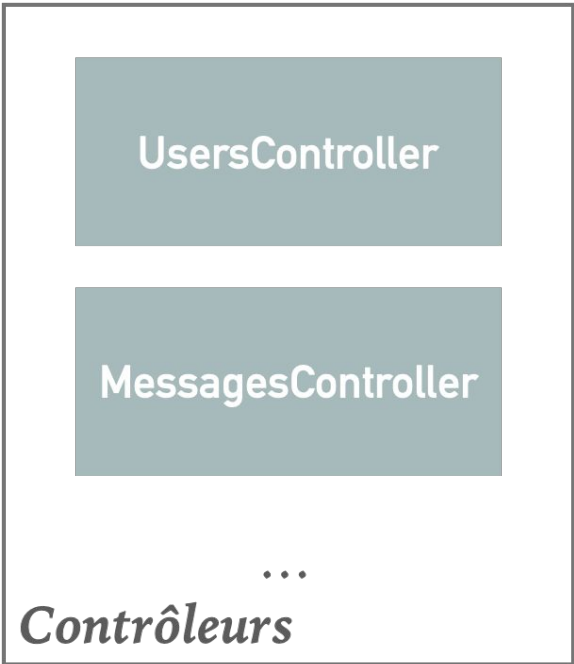
Javascript

Son / Image / Vidéo

PDF

...

Router



Base de données

1 GET /messages



Router

UserController

MessagesController

...

Contrôleurs

User

Message

...

Modèles

events/index.html.erb

messages/index.html.erb

...

Vues



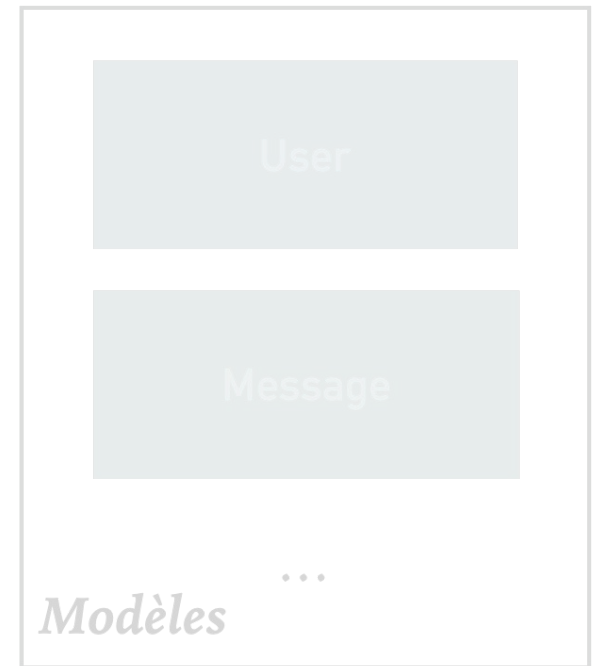
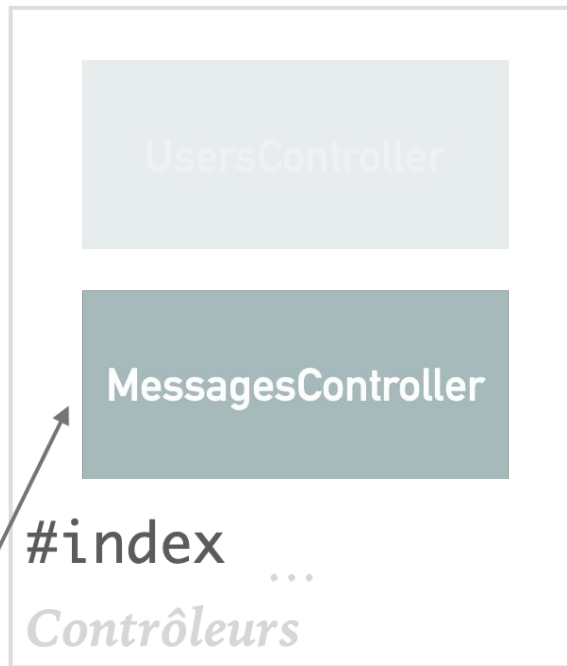
Base de données

1 GET /messages



2

#index ...
Contrôleurs



1 GET /messages



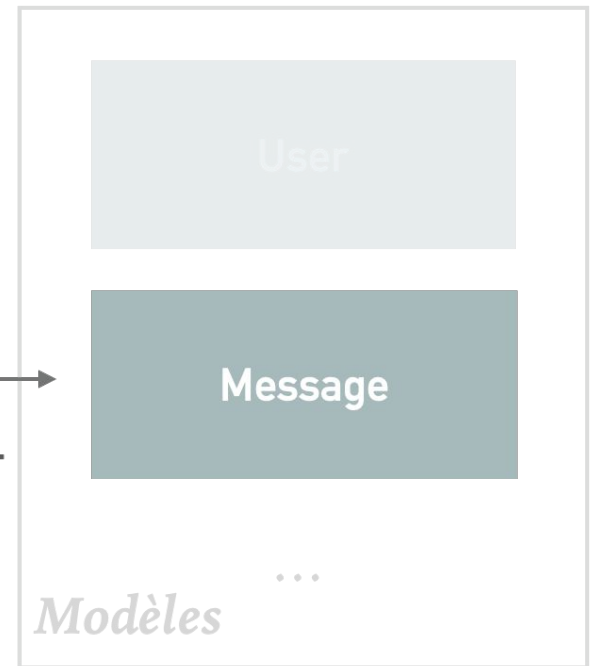
2

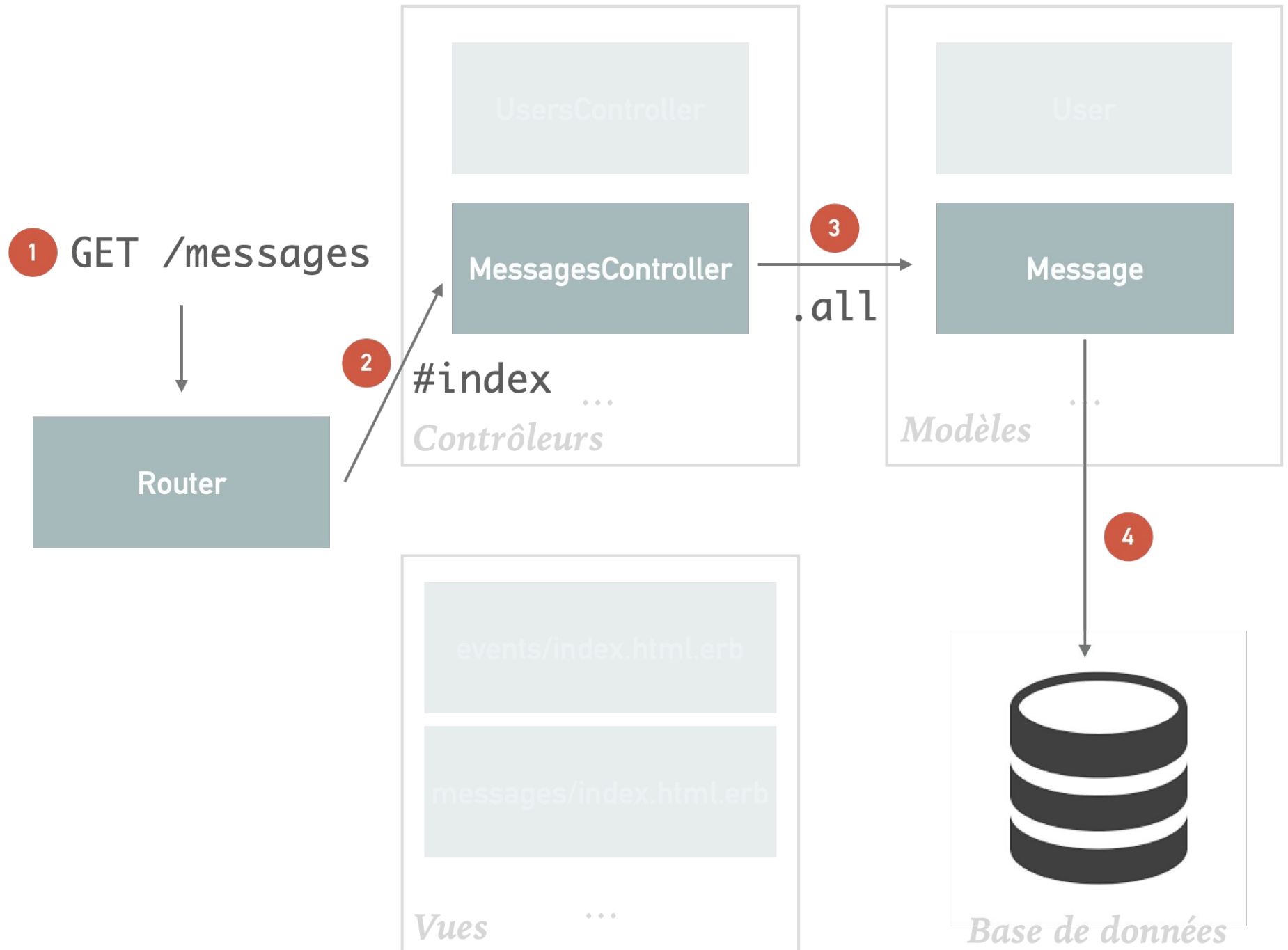
#index ...
Contrôleurs

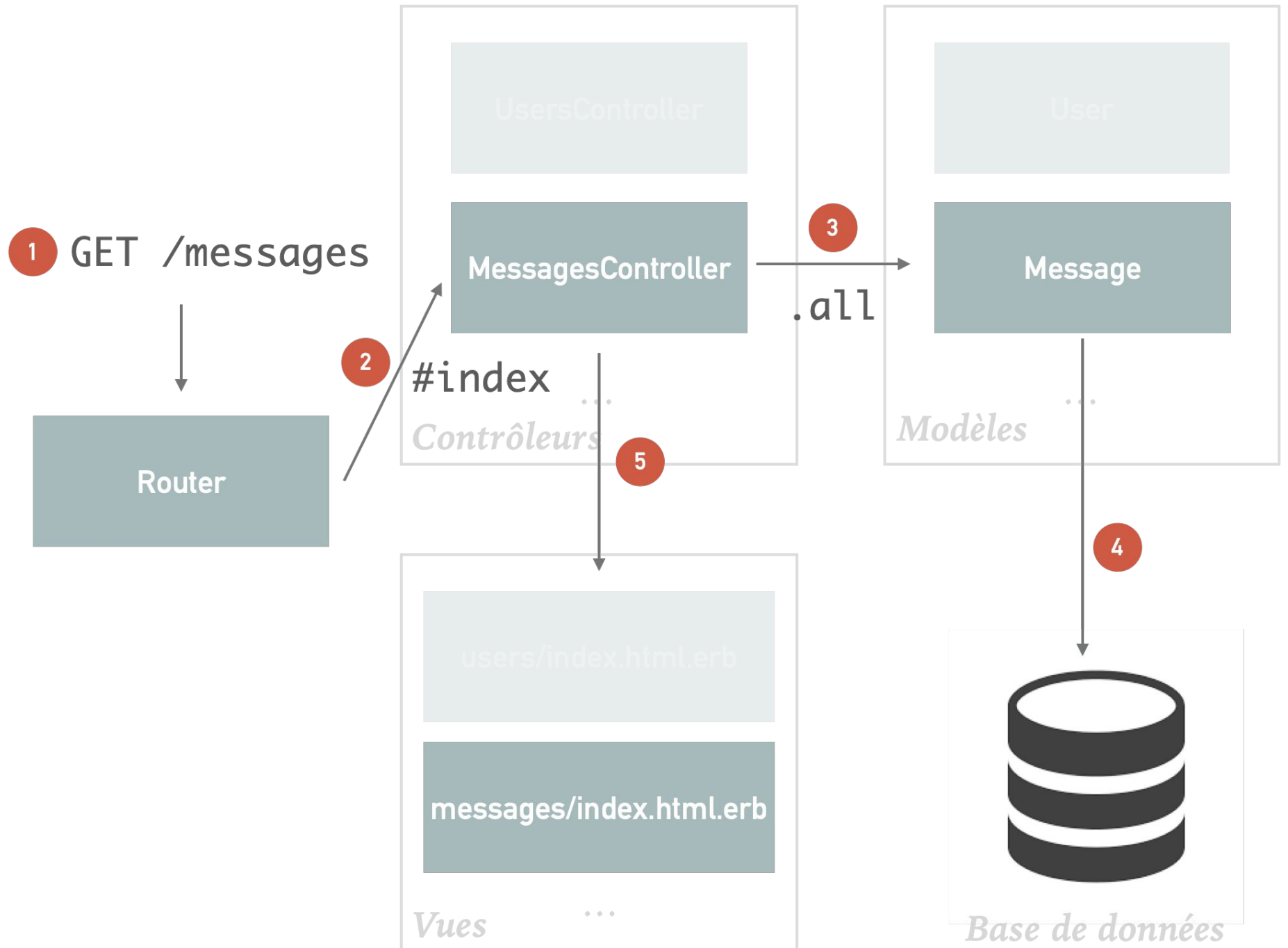


3

.all







COMMENT
COMMUNIQUER
AVEC LA
BASE DE DONNÉES

« Donne-moi tous les messages stockés en base de données, et stocke-les dans une variable `messages` »

```
messages = Message.all
```

« Crée un nouveau message et
sauvegarde le en base de données »

```
Message.create(content: "Hello RailsGirls!")
```

« Trouve le message qui possède l'identifiant **5487** en base de données »

```
message = Message.find(5487)
```

« Met à jour le contenu du message »

```
message.update(content: "J'apprends à coder")
```

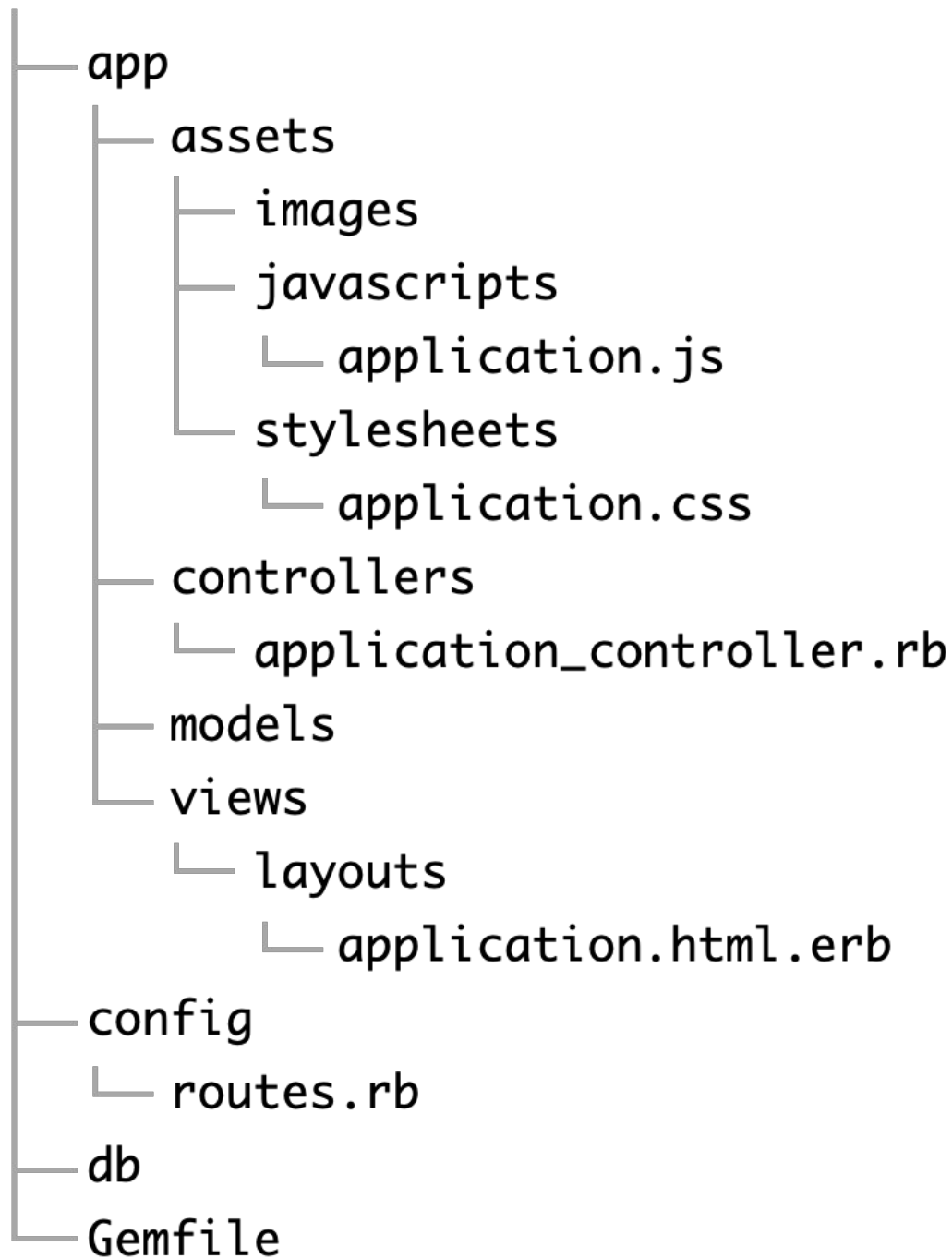
« Supprime le message »

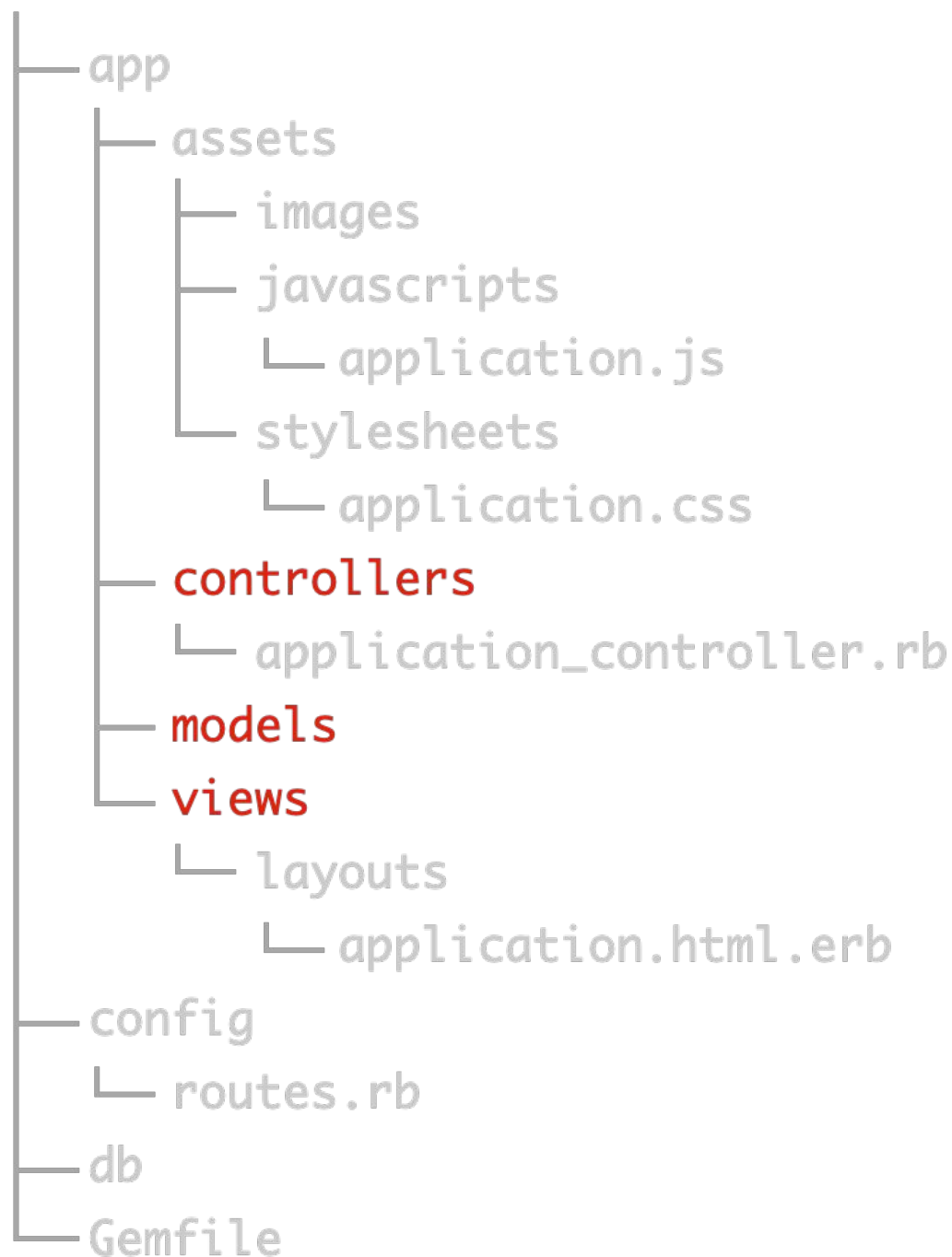
```
message.destroy
```

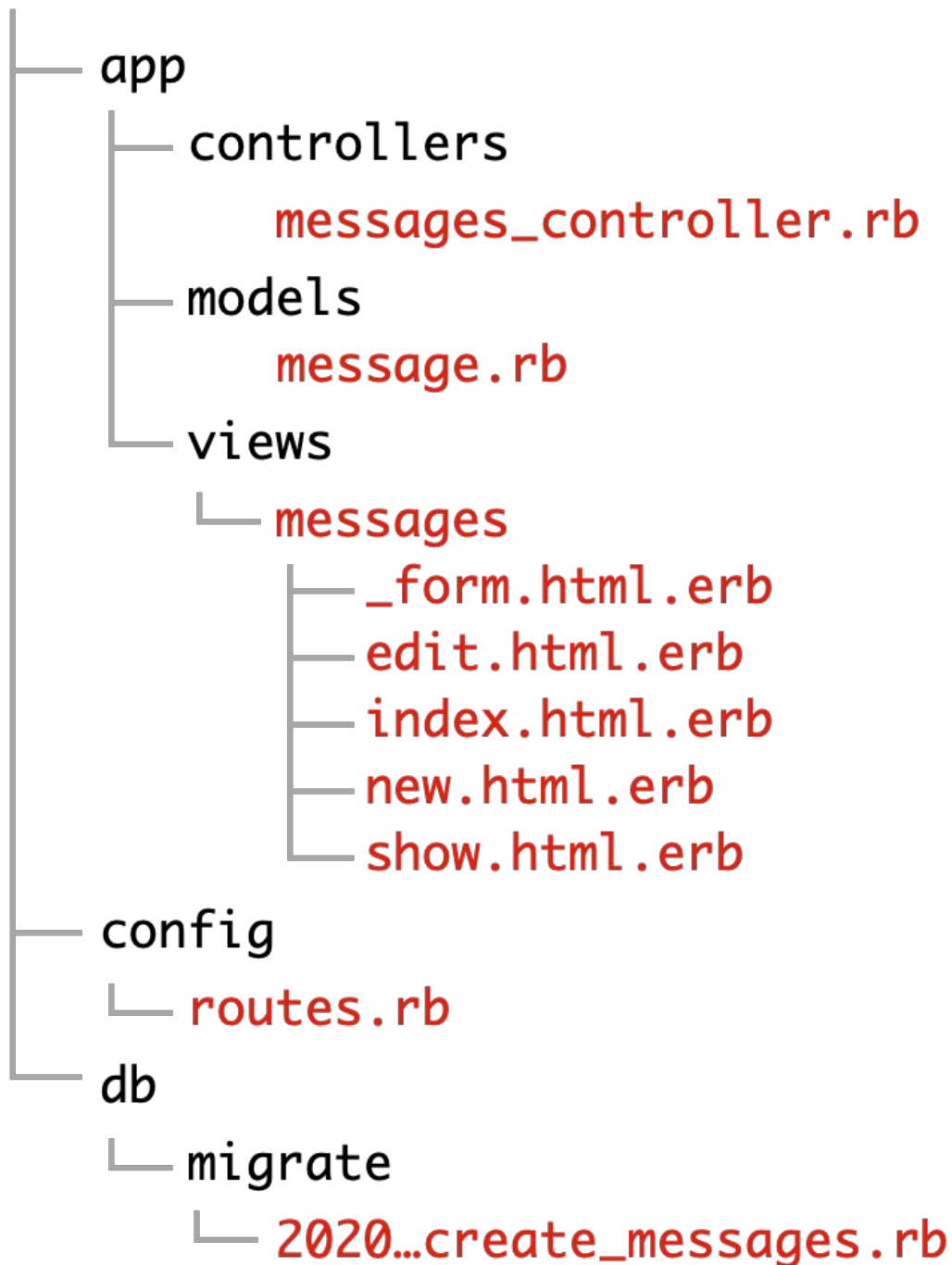
Démo !

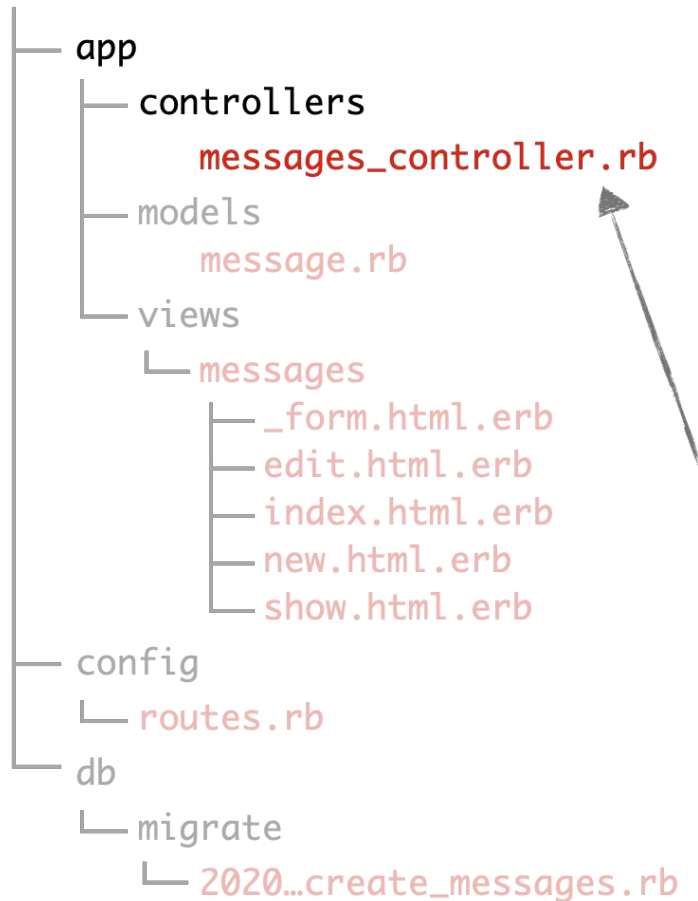
<https://github.com/nantesrb/touiteur-demo>

RÉCAP DE L'ARCHITECTURE D'UNE APP WEB RUBY ON RAILS









index *Récupération de la liste des messages*

new *Préparation à la création d'un nouveau message*

create *Création d'un nouveau message*

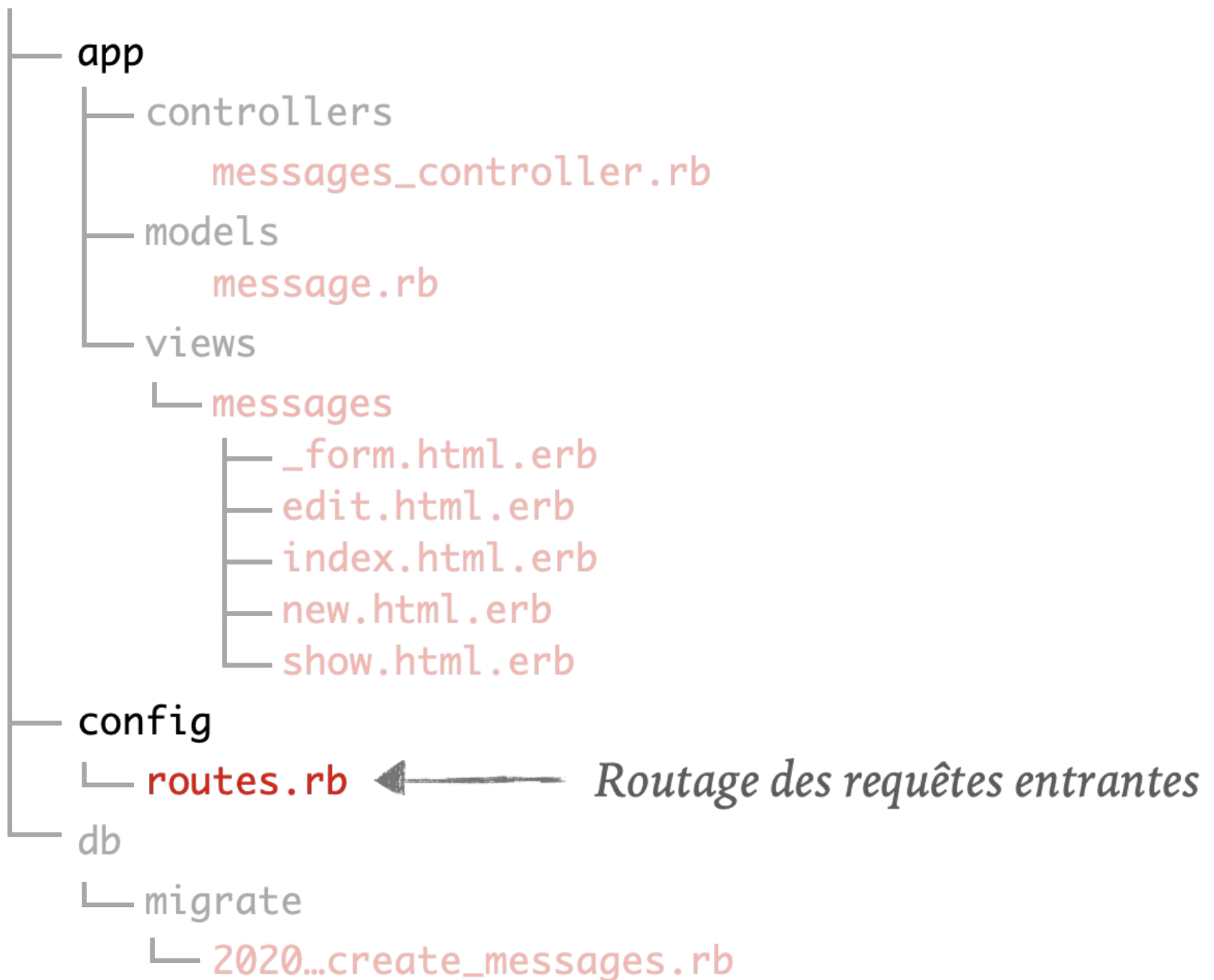
edit *Préparation à la modification d'un message existant*

update *Modification d'un message existant*

destroy *Suppression d'un message*







C'est parti !

<https://github.com/nantesrb/railsgirls-livecode-instructions>